New feature-preserving filter algorithm based on *a priori* knowledge of pixel types

Andrew H. S. Lai, MEMBER SPIE
Nelson H. C. Yung
University of Hong Kong
Department of Electrical and Electronic Engineering
Pokfulam Road, Hong Kong
E-mail: nyung@hkueee.hku.hk

Abstract. The concept and algorithmic details of a new corrupted-pixel-identification- (CPI)-based estimation filter are presented. The approach is by transforming a noisy subimage centered on a corrupted pixel into its discrete cosine transform (DCT) domain, and approximating the transformed subimage by its DC (average) coefficient only, an estimation of the noise distribution is made by combining the knowledge of the number of corrupted pixels in the subimage and the pixel intensity of the noise term. This enables the DC coefficient of the restored image in the DCT domain to be determined, and from this, the restored pixel intensity can be calculated by an inverse DCT. The whole restored image can be obtained after all the corrupted pixels are exhausted. From an extensive performance evaluation, it was found that the new algorithm has a number of desirable characteristics. First, the CPI-based estimation algorithm performs extremely well when heavily degraded images are concerned. Second, the CPI-based estimation algorithm has acceptable feature-preserving properties, far better than the conventional median filter. Third, the new algorithm can be applied iteratively to the same noisy image. Fourth, the computing speed of the CPI-based estimation algorithm is almost three times faster than the conventional median filter, and 1.6 times faster than the original CPI algorithm, making it the fastest algorithm in this class so far.

Subject terms: noise removal; feature preservation; corrupted pixel identification; impulse white noise; noise estimation; mean-square error; selective filtering.

Paper 27026 received Feb. 20, 1996; revised manuscript received July 12, 1996; accepted for publication July 13, 1996.

1 Introduction

Image filtering is commonly employed in digital image processing applications where images are degraded by randomly populated noise of the signal dependent and independent kinds.[1,2] Of these, additive random white impulses are fairly common due to poor input sampling of the image and/or the image is being interfered by an external source during transmission.[2,3] Digital image filters are designed, therefore, with the primary goal to recover the original image from the noisy image. Ideally, if the noise distribution is known exactly, the original image can be recovered from the noisy image completely in the additive case. However, it is often not possible to know exactly the nature of the noise, and the spatial randomness of the noise makes the determination of the noise content a process of estimation. As a result, the restored image resembles the original, but is never the same. In this respect, the process of image filtering becomes a matter of minimizing the errors representing the difference between the original and the restored images such that the restored image is both objectively and subjectively acceptable. A typical image degradation and restoration model is shown[2–5] in Fig. 1. Mathematically, the additive nature of the noise enables the noisy image to be represented by the summation of the original image and a noise distribution as given by Eq. (1), where $g(x,y)$ denotes the noisy image, $f(x,y)$ denotes the original image, and $\eta(x,y)$ denotes the additive noise term:

$$g(x,y)=f(x,y)+\eta(x,y). \tag{1}$$

From Eq. (1), the process of image filtering can be interpreted as given $g(x,y)$ and the *a priori* knowledge of the statistical nature of $\eta(x,y)$, an approximation, $\hat{f}(x,y)$ can be determined. If $f(x,y)$ is also known, an objective measurement may be performed, which relies on calculating the mean-square error between the original image and the restored image.[6] This is given by

$$\sigma_{\mathrm{ms}}=\frac{\Sigma_{y=1}^{H}\Sigma_{x=1}^{W}[f(x,y)-\hat{f}(x,y)]^2}{W\times H}, \tag{2}$$

where $W\times H$ is the dimension of the image.

Many noise filtering algorithms in the spatial domain have been developed. Typical examples of these filters are the median filter,[2,7–13] the averaging filter,[4,8,9] the sigma filter,[4] the box filter,[2,4] and the general rank filter.[2] These spatial filtering algorithms are mostly designed for removing a specific noise distribution. For example, the median filter is designed to remove impulse noise, while the sigma filter is designed to remove Gaussian noise. Of all these, the median filter is the most widely used for two reasons. First,

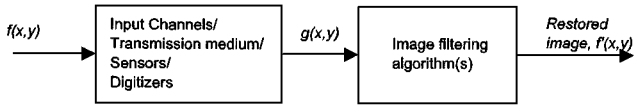


Fig. 1 Image degradation and restoration model.

it is rather effective in removing salt-and-pepper noise and similarly effective in removing white or black impulses if the SNR is high. Second, it introduces a relatively small amount of distortion to lines and edges in the image compared with some other filter algorithms. These two features are desirable, as the ideal result of image filtering is to have all the noise removed and none of the image features affected. The performance of the median filter is close to this. Similarly, the sigma filter has noise removal and feature-preserving characteristics comparable with the median filter in the case of removing Gaussian noise. In general, the sigma filter performs slightly better for images degraded by Gaussian noise, whereas the median filter performs slightly better for images degraded by impulse noise.

Furthermore, it is observed that all these filtering algorithms share one common characteristic: every single pixel in the image is subjected to the same filtering process disregarding the nature of the pixel. The reason is simply that such filtering algorithms do not have any knowledge of the nature of the pixels. In other words, they are not designed to distinguish which pixel is affected by the noise distribution and which is not. Therefore, all the pixels must be considered equally and treated in exactly the same way. This approach has two obvious effects at least. First, since the pixels unaffected by noise are also processed, certain distortion to the image content will be inevitably introduced as a result, especially when these pixels are usually in the majority. Such distortion may be unacceptable as it can reduce the sharpness of lines, edges, and boundaries. Second, processing the whole image wastes a significant amount of computing resources and may become critical in real-time applications. This problem will likely become worse as the size of image is becoming larger in multimedia processing, and the real-time demand of such applications is stringent.

In the light of this, if the nature of the pixels is known, there would be at least two advantages. First, for pixels of different natures, different treatment may be employed so that fine details in the image can be retained. In the simplest case, corrupted pixels can be processed by a filter window, while the uncorrupted pixels are not processed at all. Second, if only a selected subset of pixels is processed, substantial computing resources can be saved, given the algorithmic overhead for identifying the pixels' natures is smaller than the computing resources wasted on processing the whole image. Along the line of this argument, a generalized mean filter algorithm was developed for removing impulse noise using the concepts of thresholding and complementation.[11] This particular algorithm performs as well as the median filter, but the computing overhead required for identifying pixel types is high, and its overall delay is expected to be longer than that of the median filter.

Instead of considering every single pixel in the image, Yung and Lai[3,14] proposed a subimage approach, called the corrupted pixel identification (CPI) algorithm, which divides the whole image into subimages and identify the pixels' natures within each subimage using a threshold criterion. The CPI algorithm coupled with a median filter as its filter section (CPI-median) has been evaluated extensively and the results show that it offers a number of attractive properties. First, it is faster both in theory and practice for filter window size larger than 3. Second, it is more capable of removing black or white impulses, Gaussian, and salt-and-pepper impulses than the median, sigma, and averaging filters.[14,15] And third, it also has a good feature-preserving property that is missing in other conventional filter algorithms. The distortion introduced by the CPI-median algorithm is almost negligible compared with the other filter algorithms. However, the CPI-median algorithm fails under heavily corrupted cases. One of the reasons is that the median filtering core employed to process the corrupted pixels is unable to remove noise pixels that are clustered together. This is due to the fact that the median filter operates on the assumption that noisy pixels are in the minority within the filter window. The desire to find a CPI-based algorithm that will replace the median filter core and is capable of handling the heavily corrupted cases and yet still has an acceptable feature-preserving ability becomes the motivation of the research presented in this paper.

In this paper, the concept and algorithmic details of a new CPI-based estimation filter are presented. The philosophy of the algorithm is that if we can estimate the noise distribution within a subimage by using the *a priori* knowledge of the pixel types, then the reconstructed image within the subimage can be determined by subtracting the noise term from the noisy image. This seemingly straightforward philosophy hinges heavily on how the noise term is estimated, which is partly determined by the nature of the noise degradation and partly determined by the estimation algorithm employed. By considering the noise distribution to be strong, spikelike impulses, the issues of how the *a priori* knowledge of the pixel types from the CPI algorithm can be effectively used and the estimation of the restored image by the DC (average) coefficient of the discrete cosine transform (DCT) on a subimage are considered. The idea is simply by transforming a noisy subimage centered on a corrupted pixel into its DCT domain, and approximating the transformed subimage by its DC coefficient only, an estimation of the noise distribution is made by using the knowledge of the number of corrupted pixels in the subimage and the pixel intensity of the noise term. This enables the DC coefficient of the restored image in the DCT domain to be determined, and from this, the restored pixel intensity can be calculated by an inverse DCT. As a result, the whole restored image can be obtained after all the corrupted pixels are exhausted.

The CPI-based estimation algorithm was evaluated extensively together with the CPI-median and a conventional median filter in terms of their noise removal potential against white impulse noise, their feature-preserving ability, their effect on the restored image if they are applied iteratively, and their computing requirements. From the results of the evaluation, a number of observations can be made. First, the CPI-based estimation algorithm performs extremely well when heavily degraded images are concerned. This is the case where CPI-median algorithm fails

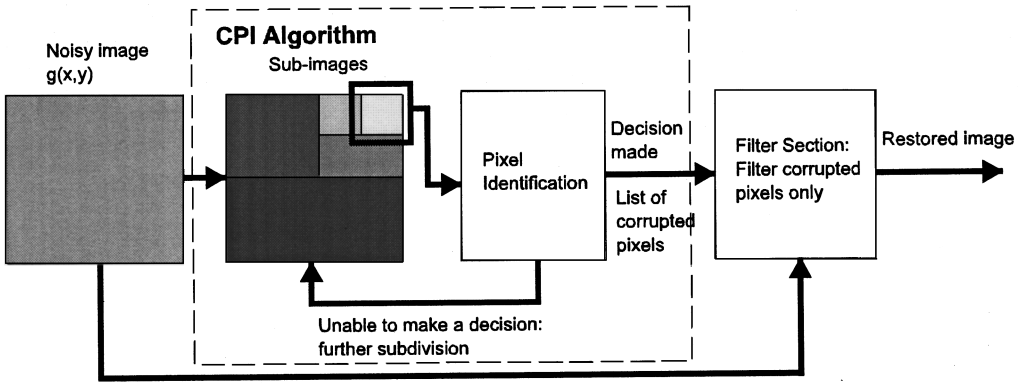


Fig. 2 CPI model.

to perform. On the other hand, the CPI-based estimation algorithm does not perform as well as the CPI-median when SNR is high. In both cases, the performance of the conventional median filter is some way behind the two CPI-based algorithms. Second, the CPI-based estimation algorithm has an acceptable feature-preserving property, although it is not as good as the CPI-median algorithm. The median filter has the worst feature-preserving performance among the three. Third, one of the characteristics of the CPI-median algorithm is that it can be applied iteratively to improve the mean-square error as well as the visual quality of the restored image when the image is heavily degraded. On this point, the median filter is not suitable to be applied iteratively at all due to the substantial distortion caused after each iteration. In the case of the CPI-based estimation algorithm, our analysis shows that using the algorithm iteratively will introduce a small degree of feature distortion each time, but the effect is not significant. However, it is best not to apply the algorithm iteratively since the majority of the noise is removed after the first iteration. Fourth, the computing speed of the CPI-based estimation algorithm is almost three times faster than the conventional median filter, and 1.6 times faster than the CPI-median algorithm, making it the fastest algorithm in this class so far.

The organization of this paper is as follows: Sec. 2 gives a brief overview of the CPI algorithm; its decision functions; its identification success rate; and its performance, merits, and drawbacks. Section 3 brings out the concept and philosophy of the CPI-based estimation filtering methodology and the algorithm in detail. Section 4 presents the performance evaluation results of the CPI-based estimation filter, focused on the capability of removing white impulse noise, feature-preserving property, iterative application property, and computing requirement of the algorithm. Section 5 concludes the merits and pitfalls of the CPI-based estimation filtering algorithm.

2 CPI Algorithm

2.1 *Pixel Identification Model*

The aims of this model are, first, to improve or maintain a noise removal capability comparable with existing conventional filter algorithms and, second, to achieve a high feature-preserving property that will enable the algorithm to be used in applications where even slight feature distortion is not tolerable. Developed on the argument of processing with discrimination, the CPI algorithm was developed on the basis that, by classifying the nature of the pixels in a noisy image, the only pixels needed to be processed are the corrupted pixels.[3,4] With this, a number of characteristics are obvious. First, the computing requirement of the algorithm will be proportional to the number of corrupted pixels identified instead of the image size. Second, image feature is preserved because of the selective filtering. Third, the accuracy of the filtering will be determined by the identification success rate (ISR). If the CPI algorithm has a low ISR, a poor restored image is to be expected. Fourth, the CPI algorithm will incur a certain amount of computation overhead that will have to be comparable with the conventional algorithms, if not better.

The model of the CPI algorithm is depicted in Fig. 2. It first divides the image $g(x,y)$ into subimages of which each subimage is tested if the difference between the maximum intensity and the minimum intensity in the subimage satisfies a condition. This condition is that if the intensity difference within the subimage is large, then no decision should be made, and further subdivision is required, otherwise the mean intensity of the subimage should be calculated and used as the threshold to classify the pixels in the subimage. The number of black and white pixels in the subimage are counted, and the group that is in the minority is considered corrupted. The lists of corrupted pixels of all the subimages are combined after the whole image has been interrogated. The combined list is used for selectively filtering the noisy image in the filter section. Note that the subimage approach is used only for generating the list of corrupted pixels. The subsequent filtering section could use any filter algorithms or window sizes on the noisy image itself.

2.2 *Decision Functions*

The decision functions for the subimage division and pixel identification of the CPI algorithm are given by equations (3) and (4). For subimage division:

Table 1 ISR of the CPI algorithm.

SNR (dB)	"Lady"	"Mickey"	"Lab"	"Casino Ticket"	"Sample"
50	70.97	70.27	67.26	75.23	93.31
30	74.80	77.48	73.90	81.04	94.04
10	82.22	86.17	81.78	86.11	95.02
0	85.51	89.54	84.56	87.20	95.36
−10	87.98	92.13	86.62	88.45	95.94
−30	91.21	94.68	89.03	88.63	96.45
−50	92.43	95.76	89.87	88.92	96.70

$$I_i(m,n)=\max_{\substack{x=0\\y=0}}^{\substack{x=m-1\\y=n-1}}[g(x+x_i,\ y+y_i)]$$

$$-\min_{\substack{x=0\\y=0}}^{\substack{x=m-1\\y=n-1}}[g(x+x_i,\ y+y_i)\},\tag{3}$$

where $I_i(m,n)$ is the intensity spread of the subimage $S_i(m,n)$, and m and n are the dimension of the subimage. If $I_i(m,n)$ is greater than a predefined maximum intensity spread and the size of the subimage $S_i(m,n)$ is greater than the minimum subimage size $S(m_0,n_0)$, then divide $S_i(m,n)$ into two equal but smaller subimages according to: If $m \geqslant n$ then $S_{i+1}(m/2,n)$, else $S_{i+1}(m,n/2)$.

For pixel identification:

$$\overline{g(x+x_i,\ y+y_i)}=\begin{cases}1 & \text{if } g(x+x_i,\ y+y_i)>M_i(m,n)\\ 0 & \text{if } g(x+x_i,\ y+y_i)\leqslant M_i(m,n)\end{cases}$$

$$\text{for } x=0,\ldots,m-1 \text{ and } y=0,\ldots,n-1,\tag{4a}$$

Corrupted pixels

$$=\begin{cases} g(x+x_i,\ y+y_i) \text{ where } \overline{g(x+x_i,\ y+y_i)}=0 \\ \quad \text{when } \displaystyle\sum_{y=0}^{n-1}\sum_{x=0}^{m-1}\overline{g(x+x_i,\ y+y_i)}\geqslant\frac{mn}{2}+\text{bias} \\ g(x+x_i,\ y+y_i) \text{ where } \overline{g(x+x_i,\ y+y_i)}=1 \\ \quad \text{when } \displaystyle\sum_{y=0}^{n-1}\sum_{x=0}^{m-1}\overline{g(x+x_i,\ y+y_i)}<\frac{mn}{2}+\text{bias},\end{cases}\tag{4b}$$

where $M_i(m,n)$ denotes the mean intensity within the subimage and the other variables have their usual definitions.[14] The bias in Eq. (4b) is determined by the type of noise corruption, which can be greater than or equal to zero for white impulse noise and less than zero for black impulse noise.

2.3 *ISR*

In the CPI filtering model, two possible scenarios could cause the identification accuracy become low. First, the originally corrupted pixels are identified as not corrupted and not filtered. Second, the originally not corrupted pixels are identified as corrupted and filtered. Neither of these two scenarios is desirable and should be minimized. Ideally, if all the corrupted and uncorrupted pixels are correctly identified (100% ISR) and the filtering operation can completely reconstruct the original uncorrupted pixels, then the restored image resembles exactly the original image. In practice, the ISR can be as high as 96% for certain type of images, using the definition given by Eq. (5). It is anticipated that such a success rate would be maintained or even improved with better decision functions. As for the filter operation, a perfect reconstruction may not be possible, and therefore, a close approximation with a small amount of error is to be tolerated. It is the purpose of this section to explore the implications of the ISR. The issue of filter operation and a new estimation filter based on the CPI algorithm are discussed in Sec. 3.

From the preceding argument, the ISR can be defined by

$$\text{ISR}=\frac{N(p_{c/c})+N(p_{\overline{c/c}})}{W\times H}\times 100,\tag{5}$$

where $p_{c/c}$ is a corrupted pixel identified as corrupted, $p_{\overline{c/c}}$ is an uncorrupted pixels identified as uncorrupted, $N(p_{c/c})$ is the number of corrupted pixel identified as corrupted; and $N(p_{\overline{c/c}})$ is the number of uncorrupted pixels identified as uncorrupted.

Extensive evaluation of the ISR with respect to the SNR and the type of images have been conducted and the results are shown[16] in Table 1. These ISR results are obtained for SNR ranges from 50 to −50 dB and over five different images. Noisy images are generated by adding random impulse white noise to the original images. The distribution and density of the noisy pixels are calculated[14] from the SNR. These images are "Lady," a head-and-shoulder picture against a window; "Mickey," a Mickey mouse key ring against a smooth background with fairly clear line and edge definitions; "Lab," a busy image of a laboratory scene with a lot of fine details; "Casino Ticket," a lottery ticket with coarse and fine objects; and "Sample," a computer-generated image with well-defined lines and shapes over a smooth background. These images are depicted in Fig. 3.

From Table 1, we can make two observations. First, in general, the ISR is inversely proportional to the SNR for all the images tested, implying the CPI algorithm performs better in the case of heavily degraded images, and not so

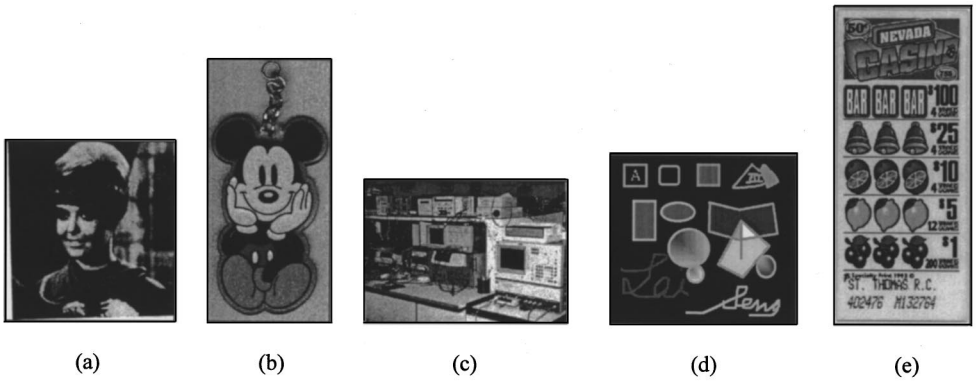


Fig. 3 (a) ''Lady,'' (b) ''Mickey,'' (c) ''Lab,'' (d) ''Sample,'' and (e) ''Casino Ticket.''

well for the lightly degraded images. As the identification decision function only conducts a binary decision, the chances of the uncorrupted image feature being identified as corrupted in the lightly degraded case would be higher compared with the heavily corrupted case. Therefore, such a result is not entirely unexpected. Second, the ISRs of the four images are not the same. For similar images such as the ''Lady'' and ''Mickey,'' their ISRs are expected to be similar. The ''Mickey'' image has an ISR about 3% better than the ''Lady'' image. This can be explained by the better line and edge definitions of the ''Mickey'' image. Furthermore, the ''Lab'' image has the worst ISR throughout the entire SNR range, which can be attributed to the amount of detail in the image being misidentified as corrupted. On the other hand, the well-defined computer-generated ''Sample'' image has the best ISR at almost all values of SNR. In conclusion, the ISR of the CPI algorithm is image sensitive, and it works better with images that have well-defined line and edge features, and less effectively if the image contains a large amount of fine details.

2.4 *Performance of the CPI-Median Algorithm*

To demonstrate the performance of the CPI algorithm, a 5×5 median filter is selected to be used in the filter section, as depicted in Fig. 2. In principle, the median filter operates according to the list of corrupted pixels supplied by the CPI algorithm. Each corrupted pixel on the list is visited and the median is calculated over all the 25 pixels with the corrupted pixel being at the center of the filter window, disregarding whether any of the neighborhood pixels are corrupted or not. Figure 4(a) depicts the original image of ''Casino Ticket,'' which has a number of well-defined features as well as some fine details at the bottom and toward the right of the image. The ISR for this image is expected to be between those of the ''Mickey'' image and ''Lab'' image. Figure 4(b) shows the noisy image corrupted by white impulses at SNR$=-50$ dB, and Figs. 4(c) and 4(d) present the restored images using the conventional median filter and the CPI-median filter, respectively. It can be seen from these two images that for a heavily degraded image, a conventional median filter fails to remove some of the noise pixels due to high local noise density. As a result, they appear as noise clusters in the restored image. Moreover, the median filter also has a detrimental effect on the fine details of the image, for example, the characters at the bottom of the image have almost disappeared because of the filtering. In the case of the CPI-median filter, there are still noise clusters left in the restored image, but the severity of such looks to be less than the median filter. Furthermore, more original image features are preserved as a result

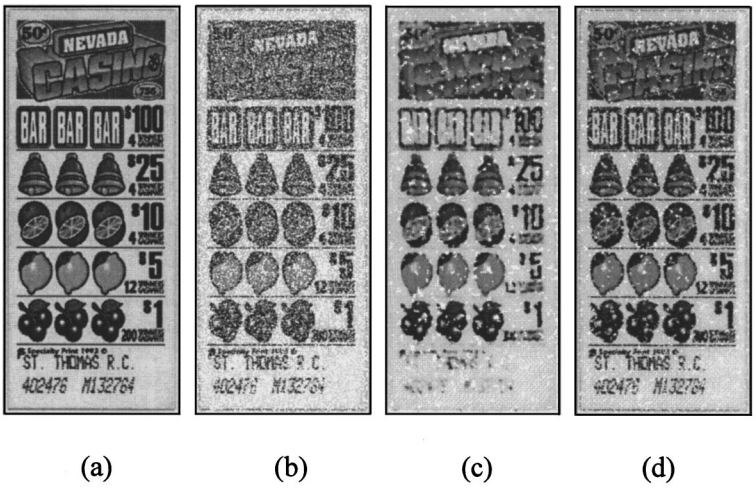


Fig. 4 (a) Original image, (b) noisy image (SNR$=-50$ dB), (c) restored by the conventional median filter, (d) and restored by the CPI algorithm with a median filter as its filter section.

of the selective nature of the algorithm. The example illustrated here agrees well with the extensive performance evaluation given in Yung and Lai[3,14] and Yung and Yung.[16]

2.5 *Merits and Drawbacks*

To simply quote the results from Yung and Lai,[14] there are a number of merits due to the CPI algorithm. First, it preserves high-frequency image features where other conventional or nonselective filters smooth out. This is a desirable property, as in applications such as video phone, the restored image is preferred to resemble the original image closely. Second, when white or black impulses or Gaussian noise is concerned, the CPI algorithm has the best noise removal performance among filters such as the median and sigma algorithms. The objective measure of the mean-square error (MSE) and the subjective visual inspection consistently highlight the superiority of the algorithm. Third, the computing resource requirement of the CPI algorithm is lower than that of the other filters both theoretically and practically. In theory, the CPI algorithm performs better than the median filter for $(2N+1)>3$. In practice, the CPI algorithm approaches two times faster than the median filter. Fourth, due to the feature preservation property of the CPI algorithm, it can be applied iteratively when the noisy image has very low SNRs. Evaluation of this particular property shows that an optimal result can be obtained after two or three iterations depending on the noise content.

However, there are a number of drawbacks associated with the CPI algorithm as well. First, if both black and white noise appear in the image, as both the black and white pixels have similar probability of being the corrupted pixels, the decision function in determining the majority pixels could wrongly identify either the black or white pixels as the uncorrupted pixels. Second, although the subimage approach is relatively fast compared with the conventional approaches, the errors due to low ISR could be significant if the SNR of the noisy image is high. Nevertheless, these errors would be translated into errors in the restored image as corrupted pixels that are not processed (insufficient noise removal), and uncorrupted pixels that are processed (feature degradation). Third, the filtering stage is performed by a conventional filter operator (median), in which the median is calculated based on all the pixels in the window, disregarding whether these neighboring pixels are corrupted or not. Since the knowledge of which pixel is corrupted is *a priori*, there is no reason why the filtering should not be carried out on a more selective basis, for example, in the median case, only the uncorrupted pixels are included in the calculation of the median. Preliminary results obtained recently indicate that substantial improvement is possible if the knowledge of pixel type is considered appropriately.[16] Section 3 essentially further explores this possibility and presents a new algorithm that uses the *a priori* knowledge. The result of combining the CPI philosophy with selective filtering is that the new algorithm gives a much better filter performance, especially for heavily degraded images.

3 CPI-Based Estimation Filter Algorithm

3.1 *Concept and Philosophy*

The philosophy of the algorithm is that if we can estimate the noise distribution using the *a priori* knowledge of the pixel types, then the reconstructed image can be determined by subtracting the noise term from the noisy image. This seemingly straightforward philosophy hinges heavily on how the noise term is estimated, which is partly determined by the nature of the noise degradation and partly determined by the estimation algorithm employed. With respect to the former, the estimation algorithm must be able to deal with the certain type of noise degradation involved, which can be accommodated in the design of the estimation algorithm. For the estimation algorithm used, its goals must be to ensure a very high percentage of the noise degradation is being removed, and equally, a very high percentage of the original features are preserved. Based on this philosophy, the following concept was developed.

By considering the noise distribution to be strong, spike-like impulses, the issues of how the *a priori* knowledge of the pixel types from the CPI algorithm can be effectively used and the estimation of the restored image by the DC coefficient of the DCT on a subimage are considered. The idea is that by transforming a noisy subimage centered on a corrupted pixel into its DCT domain, and approximating the transformed subimage by its DC coefficient only, an estimation of the noise distribution (in this case, the white impulses) is made by using the knowledge of the number of corrupted pixels in the subimage and the pixel intensity of the noise term. This enables the DC coefficient of the restored image in the DCT domain to be determined, and from this, the restored pixel intensity can be calculated by an inverse DCT. As a result, the whole restored image can be obtained after all the corrupted pixels are exhausted. The conceptual diagram of the new filter algorithm is depicted in Fig. 5.

3.2 *Algorithm*

The purpose of the DCT filter is to filter only the noisy pixel of concern (the center pixel inside the filtering window) and leave the other pixels inside the window untouched. In other words, if there are more than one noisy pixels inside the filtering window, only the noisy pixel at the center will be filtered out during the operation. Other noisy pixels are considered when they are in the center of the window. This is explained in Fig. 6. Let us define the following:

$f(x,y)$	= original image
$\eta(x,y)$	= additive noise term
$g(x,y)$	= noisy image
$\hat{f}(x,y)$	= restored image
ε	= approximated noise intensity
$F_{2N+1}(0,0)$	= DC component of $F(u,v)$ over a subimage size $(2N+1)^2$
$G_{2N+1}(0,0)$	= DC component of $G(u,v)$ over a subimage size $(2N+1)^2$
$\hat{F}_{2N+1}(0,0)$	= DC component of $\hat{F}(u,v)$ over a subimage size $(2N+1)^2$
S	= number of corrupted pixels identified by the CPI algorithm.

The discrete cosine transform of $f(x,y)$ and its inverse are given by[17,18]

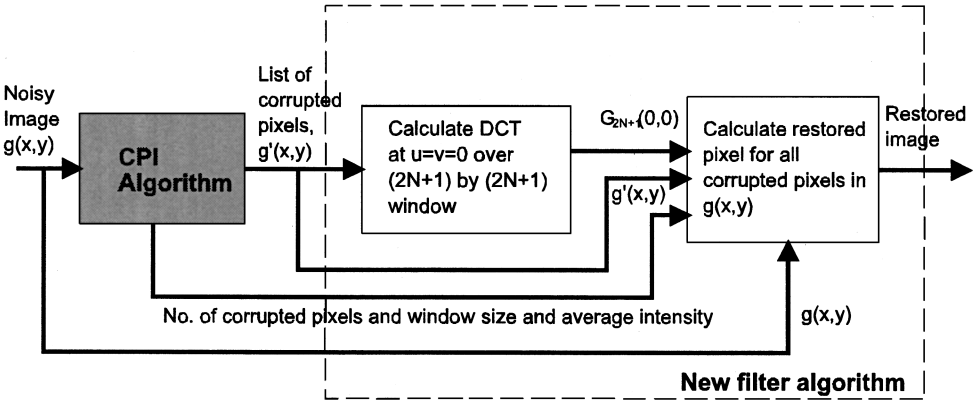


Fig. 5 Conceptual diagram of the new feature-preserving filter algorithm.

$$F(u,v)=\frac{2}{2N+1}C(u)C(v)\times\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}f(x,y)\cos\frac{(2x+1)u\pi}{2(2N+1)}\times\cos\frac{(2y+1)v\pi}{2(2N+1)}\right], \tag{6a}$$

$$f(x,y)=\frac{2}{2N+1}\left[\sum_{u=-N}^{N}\sum_{v=-N}^{N}C(u)C(v)F(u,v)\times\cos\frac{(2x+1)u\pi}{2(2N+1)}\cos\frac{(2y+1)v\pi}{2(2N+1)}\right], \tag{6b}$$

where Eq. (6a) represents the forward DCT and Eq. (6b) represents the inverse DCT. The DC component of $F(u,v)$ when $u=v=0$ over a window size of $(2N+1)^2$ is given by Eq. (7a), and the inverse is given by Eq. (7b). Equations (7a) and (7b) form a simplified DCT pair:

$$F_{2N+1}(0,0)=\frac{1}{(2N+1)}\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}f(x,y)\right], \tag{7a}$$

$$f(x,y)=\frac{F_{2N+1}(0,0)}{(2N+1)}, \tag{7b}$$

where $F_{2N+1}(u,v)$ is zero everywhere except at $u=v=0$. By taking the DCT of Eq. (1) and for $u=v=0$, the DC component of the noisy image $g(x,y)$ is given by

$$G_{2N+1}(0,0)=\frac{1}{(2N+1)}\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}g(x,y)\right]=\frac{1}{(2N+1)}\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}f(x,y)\right]+\frac{1}{(2N+1)}\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}\eta(x,y)\right]. \tag{8}$$

Assume all the additive noise pixels within the filtering window have approximately the same intensity ε, then the intensity of the noise pixels can be estimated by

$$\eta(x,y)=\begin{cases}\varepsilon & \text{if } g(x,y) \text{ is corrupted}\\ 0 & \text{otherwise.}\end{cases} \tag{9}$$

This assumption is designed to deal with the white impulse noise in question. For other types of noise distributions, Eq. (9) can be generalized to the average value or the sum of $\eta(x,y)$ over the window $(2N+1)^2$ can be approximated by the noise distribution.

With the number of corrupted pixel S being given by the CPI algorithm, we approximate the sum of $\eta(x,y)$ in Eq. (8) by $S\varepsilon$, from which $G_{2N+1}(0,0)$ can be rewritten by

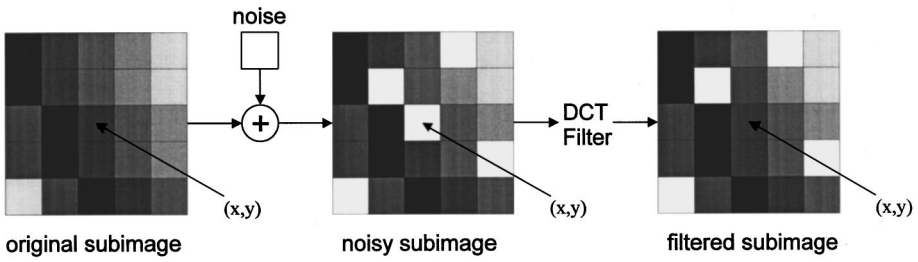


Fig. 6 Expectation of the filter algorithm.

$$G_{2N+1}(0,0)=\frac{1}{(2N+1)}\left[\sum_{x=-N}^{N}\sum_{y=-N}^{N}f(x,y)\right]+\frac{S\varepsilon}{(2N+1)}$$

$$=F_{2N+1}(0,0)+\frac{S\varepsilon}{(2N+1)}. \tag{10}$$

From Eq. (10), if the corrupted pixel at the center of the subimage is restored, then the number of corrupted pixels left in the subimage becomes $(S-1)$, and the image $f(x,y)$ is still degraded by $\eta(x,y)$ everywhere in the subimage except at the center. Therefore the DC coefficient of the DCT of $\hat{f}(x,y)$ can be represented by

$$\hat{F}_{2N+1}(0,0)=F_{2N+1}(0,0)+\frac{(S-1)\varepsilon}{(2N+1)}. \tag{11}$$

Combining Eqs. (10) and (11), we have

$$\hat{F}_{2N+1}(0,0)=G_{2N+1}(0,0)-\frac{\varepsilon}{(2N+1)}. \tag{12}$$

Let the approximated noise intensity be given by

$$\varepsilon=g(x,y)-f(x,y), \tag{13}$$

and substituting Eqs. (7b) and (13) into Eq. (10) and rearranging, we have

$$F_{2N+1}(0,0)=G_{2N+1}(0,0)-\frac{S\varepsilon}{(2N+1)}$$

$$=G_{2N+1}(0,0)-\frac{S[g(x,y)-f(x,y)]}{(2N+1)}$$

$$=G_{2N+1}(0,0)-\frac{S}{(2N+1)}\times\left[g(x,y)-\frac{F_{2N+1}(0,0)}{(2N+1)}\right]. \tag{14}$$

By further rearranging Eq. (14), we have

$$F_{2N+1}(0,0)\left[1-\frac{S}{(2N+1)^2}\right]$$

$$=G_{2N+1}(0,0)-\frac{S}{(2N+1)}g(x,y),$$

or

$$F_{2N+1}(0,0)=\frac{G_{2N+1}(0,0)-\frac{S}{(2N+1)}g(x,y)}{1-[S/(2N+1)^2]}. \tag{15}$$

By substituting Eqs. (7b), (13), and (15) into Eq. (12), the DC component of $\hat{F}_{2N+1}(0,0)$ is given by

$$\hat{F}_{2N+1}(0,0)=G_{2N+1}(0,0)-\frac{\varepsilon}{(2N+1)}$$

$$=G_{2N+1}(0,0)-\frac{1}{(2N+1)}\times\left[g(x,y)-\frac{F_{2N+1}(0,0)}{(2N+1)}\right]$$

$$=G_{2N+1}(0,0)-\frac{1}{(2N+1)}\left\{g(x,y)-\frac{1}{(2N+1)}\times\left[\frac{G_{2N+1}(0,0)-[S/(2N+1)]\,g(x,y)}{1-[S/(2N+1)^2]}\right]\right\}$$

$$=G_{2N+1}(0,0)\left[1+\frac{1}{(2N+1)^2-S}\right]-g(x,y)\times\left[\frac{2N+1}{(2N+1)^2-S}\right]. \tag{16}$$

As $g(x,y)$ is given, $G_{2N+1}(0,0)$ calculated from $g(x,y)$, and with the knowledge of S from the CPI algorithm, $\hat{F}_{2N+1}(0,0)$ can be determined. Once $\hat{F}_{2N+1}(0,0)$ is known, $\hat{f}(x,y)$ can be deduced from Eq. (7b).

4 Performance Evaluation

The purpose of this performance evaluation is to determine how well the new CPI-based estimation filter algorithm performs compared with the conventional median filter, and the CPI-median filter as described in Sec. 2. The comparison is based on evaluating the MSE between the restored image $\hat{f}(x,y)$ and the original image $f(x,y)$ in all cases where the ''Casino Ticket'' image is degraded by white impulses for SNRs ranging from -50 to $+50$ dB. The ''Casino Ticket'' consists of 256 gray levels ranging from 0 (black) to 255 (white) and having a spatial dimension of 227×533. The features of this image are that there are quite a large number of objects and characters on the ticket, the edges and lines representing the objects and characters are fairly well defined, and there are a few character fonts used such as the designed font used at the top (poor contrast), the font used for the dollar values (good contrast), and the dot-matrix printed font at the bottom of the ticket (poor print quality).

The evaluation is focused on a number of aspects:

1. noise removal capability over images degraded by white impulses
2. smoothing effect caused by the various filter algorithms
3. effect of applying these filter algorithms iteratively to the same image
4. computing resources requirement in each case.

In the evaluation, the parameter N is set to 2 in all cases, meaning the conventional median uses a 5×5 window, and the subimage size is either equal to or greater than the size of the filter to have meaningful results. Based on this argument, the CPI-median algorithm uses a 5×5 median filter and the CPI-based estimation filter algorithm also uses a

Table 2 Comparison of MSEs among different filters.

SNR (dB)	Before Filtering	Median	CPI-Median	CPI-Estimate
50	804.82	961.70	284.74	391.98
30	1866.98	1085.84	309.19	373.46
10	3664.43	1455.62	437.61	378.30
0	4708.02	1783.01	568.87	388.89
−10	5697.65	2209.70	789.32	414.33
−30	7146.99	3523.78	1532.82	473.69
−50	7863.48	4823.39	2289.47	511.27

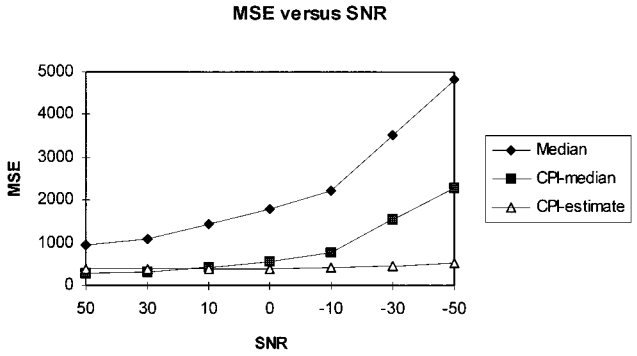


Fig. 7 MSEs of the three filters versus SNR.

5×5 subimage size for simplicity. In addition, the CPI algorithm uses a maximum intensity spread (MIS) equal to 32, which corresponds to the acceptable intensity variation of a pixel with its neighborhood pixels. For the reason for choosing these values, refer to Yung and Lai.[14]

4.1 *Noise Removal Capability*

Table 2 depicts the MSE for all the three cases, including the MSE of the noisy image before filtering. These values are plotted in Fig. 7 with the MSE versus SNR. From Fig. 7, a number of points can be observed. First, all the three filter algorithms remove noise as expected. Judging from the figures, the CPI-based filters are more effective than the median filter over the whole range. Second, the MSE behavior over the SNR range is monotonic. The MSEs of all the filters increase with decreasing SNR, with the MSE of the CPI-based estimation filter increasing fairly slightly, the CPI-median algorithm increasing at a faster rate, and the median filter increasing at the fastest rate. Third, for an SNR above 30 dB, the CPI-median performs best. For an SNR below 10 dB, the CPI-based estimation filter has the best MSE consistently.

For subjective inspection, the restored images for SNR =50, 0, and −50 dB are shown in Figs. 8, 9, and 10, respectively. In the case of a lightly degraded image (Fig. 8), the restored images of all the three filter algorithms are acceptable, perhaps, in the case of the median filter, certain lines and edges are more distorted than the other two algorithms. This can be seen at the top region where the word ''NEVADA'' is, and also the words ''BAR'' and the dot-matrix printed words at the bottom of the ticket. For the CPI-based filters, the CPI-median filter appears to have the least distortion and all the noise pixels are successfully removed. This is also true for the CPI-based estimation filter except that the restored image appears to be blurred more than the CPI-median image. This can be seen from the horizontal lines on the image. Broadly, the CPI-median image is most pleasing to look at among the three.

In the case of SNR=0 dB (Fig. 9), the restored images are still considered acceptable visually except that small isolated noise clusters can be detected in the restored images by the median filter and the CPI-median filter, on close inspection. The median filtered image has lines and edges that are obviously blurred to an extent that the characters printed at the bottom of the ticket are now totally unrecognizable. This is partly attributed to the poor print quality of the characters in the first place. On the other hand, the two CPI-based filters perform very well, with their restored images of fairly high quality. There are, however, two minor differences between the two. First, there are isolated noise clusters on the CPI-median image, but not on the CPI-based estimation filter. Second, the contrast

Fig. 8 (a) Original image, (b) noisy image (SNR=+50 dB), (c) restored by median filter, (d) restored by CPI-median filter, and (e) restored by CPI-based estimation filter.

Fig. 9 (a) Original image, (b) noisy image (SNR=0 dB), (c) restored by median filter, (d) restored by CPI-median filter, and (e) restored by CPI-based estimation filter.

of the CPI-based estimation filter seems to be a little less than the CPI-median filter. This contrast reduction is a minor feature of the CPI-type algorithms.

In the heavily degraded case as depicted in Fig. 10, the image restored by the median filter contains extensive noise clusters that makes it visually unacceptable. Its severity is such that even the well-contrasted dollar numerals on the right of the ticket are heavily distorted as well as the objects and other characters on the ticket. This poor result accounts for the large MSE for this case, as shown in Table 2. The image restored by the CPI-median filter also has a large amount of noise clusters, but the clusters are noticeably smaller and not as intense as in the median case. Nevertheless, the existence of the noise clusters in the restored image makes it visually unacceptable. On the other hand, the features of the ticket are still very much intact, with distortion mainly due to the presence of noise clusters. In the case of the CPI-based estimation filter, the restored image is virtually free of noise clusters and the lines and edges are well preserved, apart from the minor distortion that is still evident on close inspection. This makes the CPI-based estimation image the most acceptable visually, and its low MSE compared with the other two algorithms also indicates its superior performance when low SNR is concerned.

4.2 *Feature Preserving Property*

This evaluation aims to identify how well a filter algorithm preserves features when undertaking the noise filtering process. The three different filters: conventional median, CPI-median, and CPI-based estimate filters are applied to the original image as depicted in Fig. 11(a). From the MSE measured and the visual inspection of the restored images, the feature preserving property of each filter algorithm can be studied. The restored images are depicted in Figs. 11(b) to 11(d). Their absolute and relative MSEs are tabulated in Table 3.

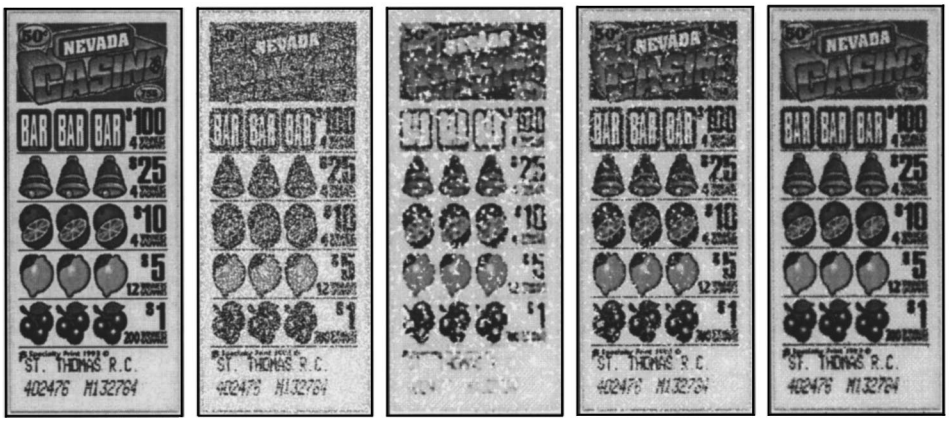


Fig. 10 (a) Original image, (b) noisy image (SNR=−50 dB), (c) restored by median filter, (d) restored by CPI-median filter, and (e) restored by CPI-based estimation filter.

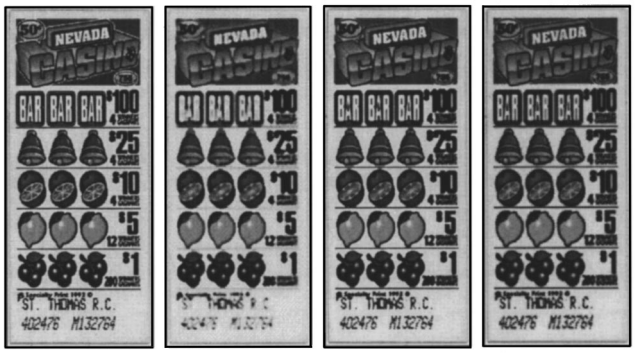


Fig. 11 (a) Original image, (b) restored by median filter, (c) restored by CPI-median algorithm, and (d) restored by CPI-based estimation algorithm.

From Table 3, it can be seen that the CPI-median algorithm gives the lowest MSE compared with the CPI-based estimation algorithm, whose MSE is 1.47 times higher, and the conventional median algorithm, whose MSE is 3.22 times higher. These values agree with the results presented in the previous sections that the CPI-median offers the lowest MSE at high SNR, followed by the CPI-based estimation. The median filter is the worst performer in such an SNR range. On visual inspection, the CPI-median image appears to resemble the original image closely with almost all features preserved. The CPI-based estimation image appears to have fine blurring at the lines and edges of objects and characters, and the dynamic range seems to be reduced more. Note, however, that these effects are minor and as a whole, most features are reasonably preserved. For the median image, blurring can be obviously seen across the whole image. In particular, the characters at the bottom of the ticket are severely distorted, the object outlines are blurred and some of the fine features are now merged together with their adjacent features.

4.3 *Iterative Application of the Filter Algorithms*

The purpose of this evaluation is to develop an understanding of how each filter algorithm performs if it is applied iteratively to the same noisy image. Although noise filters are not normally used in this way, iterative application may be useful if the image is heavily degraded. In this case, the noisy ''Casino Ticket'' images degraded at SNR$=-10$ and -50 dB were subjected to repeated application of the same algorithm for up to four times. Their MSE values are tabulated in Tables 4 and 5, respectively.

From Table 4 and Fig. 12, it can be seen that the MSE for the median filter rises slowly as the number of iterations increases. At SNR$=-10$ dB, this result is expected, as most of the noise would have been removed after the first iteration. Further application of the same algorithm will only introduce more blurring and distortion to the restored image. In the CPI-median case, the MSE has a minimum after the second iteration, and it rises very slowly beyond that. This can be explained by the fact that most of the remaining noise pixels/clusters are removed after the second iteration, after which, there is no more significant noise removal other than slight feature distortion. For the CPI-

Table 3 Comparison of absolute and relative MSEs among different filters.

	Median	CPI-Median	CPI-Estimate
Absolute	920.1	285.57	421.21
Relative	3.22	1	1.47

Table 4 Iterative performance of different filters on noisy image at SNR$=-10$ dB.

Iterations	Median	CPI-Median	CPI-Estimate
$\times 1$	2209.70	789.32	414.33
$\times 2$	2223.68	589.08	706.66
$\times 3$	2346.94	589.09	964.55
$\times 4$	2444.32	634.75	1217.13

Table 5 Iterative performance of different filters on noisy image at SNR$=-50$ dB.

Iterations	Median	CPI-Median	CPI-Estimate
$\times 1$	4823.39	2289.47	511.27
$\times 2$	4150.64	1034.94	767.17
$\times 3$	4026.85	848.39	983.51
$\times 4$	4028.97	817.84	1203.06

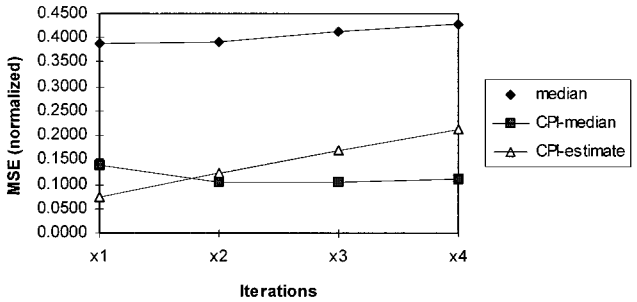


Fig. 12 Iterative performance (normalized) of different filters on noisy image at SNR$=-10$ dB.

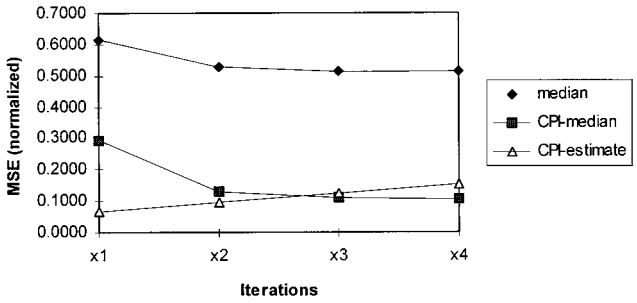
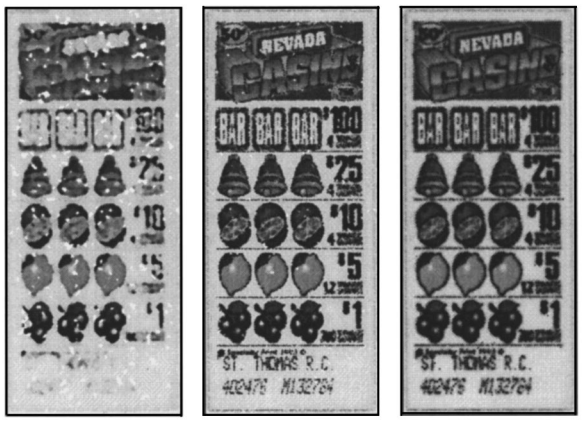


Fig. 13 Iterative performance (normalized) of different filters on noisy image at SNR=−50 dB.

based estimation algorithm, the initial MSE is the lowest. But when applied iteratively, its MSE increases rapidly above the CPI-median MSE. This can be explained by the fact that the CPI-based estimation algorithm effectively removed most of the noise after the first iteration. Further iterations simply cause more features in the image to be lost, resulting in a higher MSE.

As depicted in Table 5 and Fig. 13, the MSE behavior of the median filter is different when the SNR is low (−50 dB). Instead of rising, the MSE decreases as the number of iterations increases. This is because of the extensive noise clusters that still remain in the image after the first iteration, which are further removed in the subsequent iterations. Visually, this can be seen in Figs. 14 to 17. The number of noise clusters is reduced noticeably after a number of iterations, but unfortunately, the degree of distortion has increased to such an extent that the restored image is severely blurred [Fig. 17(a)]. In the case of the CPI-median, the MSE decreases as the number of iterations increases. The reduction is quite substantial by almost three times after the fourth iteration. This indicates that the extensive noise clusters remaining in the image after the first iteration are being removed further. A minimum in this MSE curve is expected at a higher number of iterations, after which the MSE will increase as in Fig. 12. From the images restored by the CPI-median as shown in Figs. 14 to 17, it can be

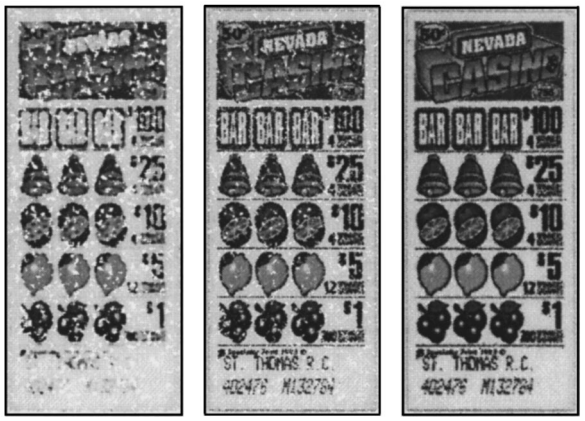


(a) Median (b) CPI-median (c) CPI-estimate

Fig. 15 Second iteration.

deduced that the CPI-median algorithm gives the best restored image at the fourth iteration [Fig. 17(b)]. Essentially, almost all noise clusters are removed from the image and most of the image features are preserved.

For the CPI-based estimation algorithm, the MSE results are on a par with the previous cases, where its MSE is better after one or two iterations, and rises to just slightly above the CPI-median's MSE at the fourth iteration. The general trend agrees with the previous case and can be explained by the fact that the CPI-based estimation algorithm is a lot more effective in noise removal and therefore gives the best MSE after just one iteration. Visually, the restored image also appears to be the best [Fig. 14(c)]. When the algorithm is applied repeatedly, there are no more noise pixels to be removed, and instead, the image features are being blurred due to its slightly poorer feature-preserving ability than the CPI-median algorithm. This accounts for the increase in the MSE over the number of iterations. It can also be seen visually that the images restored by the CPI-based estimation algorithm have deteriorating visual



(a) Median (b) CPI-median (c)CPI-estimate

Fig. 14 First iteration.

(a) Median (b) CPI-median (c) CPI-estimate

Fig. 16 Third iteration.

(a) Median (b) CPI-median (c) CPI-estimate

Fig. 17 Fourth iteration.

quality as the number of iterations increases. The extent of blurring is not as severe as the median images [Fig. 17(c)].

4.4 *Computing Resources*

It has been proven that CPI-median filter is 1.6 times faster than the conventional median filter[14] in practice. This significant improvement makes the CPI-based filtering algorithms more likely to be used in real time. In addition to this basic improvement, the CPI-based estimation filter employs a set of equations that is expected to perform even faster than the CPI-median filter. The computing times required to filter the images given in this paper by different filters are tabulated in Table 6. The hardware platform used is a Silicon Graphic workstation (Indy).

Obviously, the conventional median filter takes the same computing time for all the images because its computing requirement depends on the size of the image and the filter operation. Both of these parameters are constant, disregarding the severity of noise degradation. For the CPI-median filter, the results improve on the performance presented before, which is around two times faster. This is attributed to the different hardware platforms used. For the CPI-based estimation filter, the average time taken is around 1.46 s, which is almost three times faster than the median filter. Such comparison indicates the superiority of the CPI-based filters over the conventional filters.

5 Conclusion

Generally, the CPI-based algorithms outperform the conventional median filter algorithm in terms of objective MSE measurement, subjective visual inspection, and computing speed. Although the CPI approach appears to be more complex, the eventual outcome is attractive as algorithms based on the CPI concept have a better noise removal capability than conventional filters and can be applied iteratively to improve the MSE as well as visual quality. In addition, their feature-preserving property and faster computing speed are desirable characteristics in areas such as multimedia processing.

In particular, the CPI-based estimation algorithm has its own special features and properties. From the results of the performance evaluation given in Sec. 4.1, it can be noted that its noise removal capability is more effective for images heavily corrupted by white impulses, whereas the CPI-median filter is more suitable for lightly corrupted images. On the other hand, the CPI-based estimation filter possesses a less effective feature preserving property compared with the CPI-median filter. This is clearly demonstrated in Secs. 4.2 and 4.3, where its restored images exhibit more blurring and line/edge distortion than the CPI-median filter. However, it is fair to note that the feature-preserving ability of the CPI-based estimation filter is much higher than the conventional median filter. Although the median filter is often considered as introducing the least amount of distortion to the restored image compared with filters such as averaging and other rank filters, its feature-preserving ability is still a long way from that of the CPI-based filters. In terms of computing speed, as given in Sec. 4.4, the CPI-median filter has been reported to be 1.6 times faster than the median filter before, and the further improvement to 2 times faster simply reinforces this point. On this point, the CPI-based estimation filter is almost three times faster than the median filter, making it the fastest filter in this class with a very acceptable performance both objectively and visually. Regarding the iterative application of the CPI-based estimation algorithms, our analysis in Sec. 4.3 shows that it is indeed not necessary for the CPI-based estimation algorithm to be applied iteratively as the restored image after the first iteration is often good enough. Further iterations will cause feature degradation in the image. In contrast to this, the CPI-median filter is not as effective for heavily degraded images, and therefore, further iterations help to further remove noise. Due to its higher feature preserving ability, higher numbers of iterations do not have much impact to the image features.

Regarding future directions, a number of aspects will be studied. First, Eq. (10) will be investigated as to how different noise distributions can be represented by ε. Currently, ε is treated as a constant to represent the white impulses. If another type of noise distribution, such as Gaussian white, is to be considered, ε may represent a function that is to be determined by the noise distribution. It would be interesting to identify how this can be incorporated into the estimation of $\hat{F}_{2N+1}(0,0)$ and subsequently $\hat{f}(x,y)$. Second, as presented in Secs. 2.2 and 2.3, the identification success rate is a measure of how successful both the subimage and pixel identification decision functions are. With possible variants in the CPI-based algorithms, further investigation into the variations of the decision

Table 6 Computing times required for different filters.

SNR (dB)	Median Filter	CPI-Median	CPI-Estimate
50	4.7 s	2.5 s	1.6 s
30	4.7 s	2.1 s	1.4 s
10	4.7 s	2.5 s	1.6 s
0	4.7 s	2.2 s	1.4 s
−10	4.7 s	2.1 s	1.3 s
−30	4.7 s	2.2 s	1.4 s
−50	4.7 s	2.4 s	1.5 s

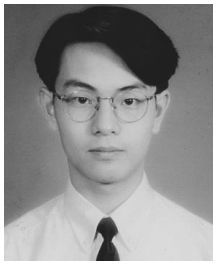
functions, or a completely new set of decision functions could point to more effective noise removal algorithms with high ISR.

Acknowledgments

The authors gratefully acknowledge the financial support of the CRCG of The University of Hong Kong under grant number 337/062/0016.

References

1. C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, pp. 198–205, Prentice-Hall, Englewood Cliffs, NJ (1992).
2. J. C. Russ, *The Image Processing Handbook*, 2nd ed., IEEE Press, New York (1995).
3. H. C. Yung and H. S. Lai, ''Novel filter algorithm for removing impulse noise in digital images,'' *Proc. SPIE* **2501**, 210–220 (1995).
4. Z. Yi, ''Image analysis, modeling, enhancement, restoration, feature extraction and their application in nondestructive evaluation and radio astronomy,'' PhD Thesis, Dept. of Electrical Engineering, Iowa State Univ. (1987).
5. A. K. Katsaggelos, *Digital Image Restoration*, Springer-Verlag, Berlin (1991).
6. R. Parsons, *Statistical Analysis: A Decision-Making Approach*, Harper & Row, New York (1978).
7. M. Sonka, V. Hlavac, and Roger Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall Computing, London (1994).
8. W. Niblack, *An Introduction to Digital Image Processing*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ (1986).
9. R. J. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York (1989).
10. Y. H. Lee and S. A. Kassam, ''Generalized median filtering and related nonlinear filtering techniques,'' *IEEE Trans. Acoust. Speech Signal Process*. **33**, 672–683 (June 1985).
11. A. Kundu, S. K. Mitra, and P. P. Vaidyanathan, ''Application of two-dimensional generalized mean filtering for removal of impulse noises from images,'' *IEEE Trans. Acoust. Speech Signal Process.* **32** (3), 600–609 (1984).
12. I. Pitas and A. N. Venetsanopoulos, ''Nonlinear mean filters in image processing,'' *IEEE Trans. Acoust. Speech Signal Process*. **34**(3), 573–584 (1986).
13. H. Hwang and R. A. Haddad, ''Adaptive median filters: new algorithms and results,'' *IEEE Trans. Image Process.* **4**(4), 499–502 (1995).
14. H. C. Yung and H. S. Lai, ''Performance evaluation of a feature preserving filtering algorithm for removing additive random noise in digital images,'' *Opt. Eng.* **35**(7), 1871–1885 (1996).
15. H. C. Yung, H. S. Lai, and K. M. Poon, ''Modified CPI filter algorithm for removing salt-and-pepper noise in digital images,'' in *Visual Communications and Image Processing'96, Proc. SPIE* **2727**, 1439–1449 (1996).
16. H. C. Yung and C. F. Yung, ''Performance evaluation of a class of CPI algorithms,'' Research Report, Dept. of Electrical and Electronic Eng., The University of Hong Kong (Nov. 1995).
17. A. Leger, ''JPEG still picture compression algorithm,'' *Opt. Eng.* **30**(7), 947–954 (1991).
18. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA (1992).



Andrew H. S. Lai received the BEng degree in computer engineering from the University of Hong Kong in 1994 and his MSc degree from the University of Surrey, United Kingdom, and was awarded the Cable & Wireless Prize for being the best overall student in the course in 1995. He is currently a PhD candidate in the Department of Electrical and Electronic Engineering, the University of Hong Kong. His research interests are digital image processing and machine navigation.

Nelson H. C. Yung received his BSc and PhD degrees in 1982 and 1985, respectively, from the University of Newcastle Upon Tyne, England, where he was a lecturer from 1985 to 1990. During this period, he led a number of large-scale research projects on image processing. From 1990 to 1993, Dr. Yung was employed by the Defence Science and Technology Organisation, Australia. He was chiefly responsible for the theoretical research and algorithmic development of military-grade signal analysis systems. He joined the University of Hong Kong in late 1993 as a lecturer. Dr. Yung has coauthored a computer vision book and a number of book chapters and has published over 50 journal and conference papers in the areas of system methodology and image processing. He has acted as referee for the *IEE Proceedings*, the *HKIE Transactions*, and a number of international conferences. He is a chartered electrical engineer and a member of IEE and a senior member of IEEE.